

# A Semi-distributed Reputation-based Intrusion Detection System for Mobile Adhoc Networks

Animesh Kr Trivedi<sup>1</sup>, Rajan Arora<sup>1</sup>, Rishi Kapoor<sup>1</sup>, Sudip Sanyal<sup>1</sup> and Sugata Sanyal<sup>2</sup>

<sup>1</sup>Indian Institute of Information Technology, Deoghat, Jhalwa, Allahabad (U.P.), India  
{aktrivedi\_b03, rarora\_b03, rkapoor\_b03, ssanyal}@iiita.ac.in

<sup>2</sup>School of Technology and Computer Science, Tata Institute of Fundamental Research, India  
sanyal@tifr.res.in

**Abstract**—A *Mobile Adhoc Network* (MANET) is a cooperative engagement of a collection of mobile nodes without any centralized access point. The underlying concept of coordination among nodes in a cooperative MANET has induced in them a vulnerability to attacks due to issues like dynamically changing network topology, cooperative algorithms and lack of centralized monitoring point. We propose a semi-distributed approach towards a reputation-based *Intrusion Detection System* (IDS) that combines with the Dynamic Source Routing (DSR) protocol for strengthening the defense of a MANET. Our system inherits the features of reputation from human behavior, hence making the IDS socially inspired. It has a semi-distributed architecture as the critical observations of the system are neither spread globally nor restricted locally. The system assigns maximum priority to self observation by nodes for updating any reputation parameters, thus avoiding the need of a trust relationship between nodes. Our system is also unique in the sense that it features the concepts of *Redemption* and *Fading* with a robust *Path Manager* and *Monitor* system. Simulation studies show that DSR fortified with our system outperforms normal DSR in terms of the packet delivery ratio and routing overhead even when up to half of nodes in the network behave as malicious. Various parameters introduced such as timing window size, reputation update values, congestion parameter and other thresholds have been optimized over several simulation runs. By combining the semi-distributed architecture and other design essentials like path manager, monitor module, redemption and fading concepts, our system proves to be robust enough to counter most common attacks in MANETs.

**Keywords:** *Adhoc networking, Security, promiscuous mode, Reputation based Intrusion Detection System*

## I. INTRODUCTION

The term *ad hoc networks* dates back to the 1970's where an ad hoc network was first setup as a part of certain defense research projects. With advances in microelectronics technology and networking protocols, it has been possible to integrate mobile nodes and various other network devices into a single unit called an *ad hoc* node. Further, interconnection of these nodes wirelessly is termed as an *ad hoc network*.

MANETs are different from conventional networks. A MANET is formed by an autonomous system of mobile nodes that are self-configuring and have no constraints, such as a fixed infrastructure or a central administration system. Nodes in MANETs are both routers and terminals. They are dynamic in the sense that each node is free to join and leave the network in a nondeterministic way. In addition, they do not have a clearly defined physical boundary, and therefore, no specific entry or

exit point. Such a network can thus be rapidly deployed and can provide the amount of flexibility and adaptability which is otherwise unattainable under adverse circumstances. Although MANET is a very promising technology, challenges are slowing its development and deployment. Nodes in ad hoc networks are in general limited in battery power, memory and CPU capacity. Hence the transmission ranges of these devices are also limited and nodes have to rely on neighbor nodes in the network to route the packet to its destination. They are sometimes referred to as multihop networks, where a hop is a direct link between two nodes. Ad hoc networks have found applications in emergency rescues, battlefield operations, mobile conferencing, national crisis, home and community networking, disaster recovery etc.

The flexible structure and volatile environment of MANETs results in significant node misbehavior. Not only does it degrade the overall network performance, but, it also becomes difficult to detect intruders on grounds of mobility and vulnerability of the nodes. Thus, there is a serious need for a robust IDS for MANETs.

Some fundamental problems of MANETs must be kept in mind while designing any security solution. Firstly, it is often very hard to differentiate intrusions and normal operations or conditions in MANETs because of the dynamically changing topology and volatile physical environment. Secondly, mobile nodes are autonomous units that are capable of roaming independently in an unrestricted geographical topology. This means that nodes with inadequate physical protection can be captured, compromised or hijacked. Thirdly, decision-making in ad hoc networks is usually decentralized and many ad hoc network algorithms rely on the cooperative participation of all nodes. Most ad hoc routing protocols are also cooperative in nature and hence can be easily misguided by false routing information. Without any counter policy, the effects of misbehavior have been shown to dramatically decrease network performance. In this paper, we propose a new technique based on reputation for efficiently solving the problem of intrusion detection.

The next section gives a brief background about routing related issues in MANETs, section III entails a discussion of some related efforts which is followed by the system design overview in section IV. Section V describes the protocol and

the following section VI talks about its implementation details. Simulation results and optimization procedures for parameters such as window size are given in the section VII. The last section presents some concluding remarks.

## II. BACKGROUND

In order to understand the nature of attacks on MANETs, we first need to look at the routing protocols for these networks. They have been classified under two main categories - Proactive and Reactive routing protocols. *Proactive* protocols work with tables that are used to store routing information and updates are triggered to propagate any information about changes throughout the whole network. The obvious advantage is that routes to any destination node are always available without the overhead of generating a *route request* whenever the need for a route arises. But, an extra overhead is always a major issue before deploying a proactive routing protocol, because it generally affects the overall throughput and power usage. Destination-Sequenced Distance Vector (DSDV) [1], Wireless Routing Protocol (WRP) [2], Cluster Gateway Switch Routing (CGSR) [3] are some common examples.

On the other hand, *Reactive* routing protocols are *on-demand* i.e. a route discovery mechanism is initiated whenever there is a need for setting up a path for communication between a source and a destination node. The source node initiates route discovery by flooding the network successively with route queries. The destination node on receiving a route request (RREQ) addressed to it, sends back a route reply (RREP) message as unicast to the source node either through the discovered route or by initiating another route request. Generally, on-demand routing requires less overhead than table-driven routing; but it incurs a path discovery delay whenever a new path is needed. Dynamic Source Routing protocol (DSR) [4], Adhoc On-Demand Distance Vector Routing (AODV) protocol [5], Temporally Ordered Routing Algorithm (TORA) [6], Associativity-Based Routing (ABR) [7], Signal Stability Routing (SSR) [8], Zone-Based Hierarchical Link State Routing Protocol (ZRP) [9] are a few examples.

Attacks are possible on reactive protocols like DSR due to lack of built-in security measures and the assumption of honest coordination and cooperation among nodes and with the protocol. We will outline a few attacks by nodes below, the others are discussed in detail by Sonja Buchegger et. al. [10]:

- Dropping all packets not destined to it or performing only partial dropping. Partial dropping can be restricted to specific types, such as only data packets or route control packets or packets destined to specific nodes.
- Sending forged routing packets, an attacker can create a so-called black hole, a node where all packets are discarded or all packets are lost.
- Modifying the nodes list in the header of a RREQ or a RREP to misroute packets and adding incorrect routes in

the route cache of other nodes.

- Decreasing the hop count DSR(TTL) when receiving a packet, so that the packet will never be received by the destination. This attack could be detected by the previous node in route by enhanced passive acknowledgment.
- Initiating frequent RREQ to consume bandwidth and energy and to cause congestion.

## III. RELATED WORK

Reputation-based systems are a new paradigm and are being used for enhancing security in different areas. These systems are lightweight, easy to use and are capable of facing a wide variety of attacks. Among these mechanisms, CORE [11], CONFIDANT [12] and OCEAN [13] gain a special mention.

Reputation based systems do not rely on the conventional use of a common secret to establish confidential and secure communication between two parties. Instead, they are simply based on each other's observations. Reputation based systems are used for enhancing security in adhoc networks as they model cooperation between the nodes which is inspired from social behavior. Such systems are used to decide whom to trust and to encourage trustworthy behavior. Resnick and Zeckhauser [14] identify three goals for reputation systems:

- To provide information to distinguish between a trustworthy principal and an untrustworthy principal.
- To encourage principals to act in a trustworthy manner
- To discourage untrustworthy principals from participating in the service the reputation mechanism is present to protect.

*Watchdog* and *Path-rater* [10] are some essential components of any typical reputation based IDS. Watchdog performs the activity of monitoring its neighborhood and based on these observations, pathrater ranks the available path in route cache. Misbehavior detection and reputation-based intrusion detection may be either distributed or local. Here, fully distributed means that information regarding one's reputation change is immediately propagated in the whole network. In the latter case, called local reputation based systems, nodes are fully dependent on their personal opinion about other nodes' reputation and behavior.

Distributed IDS protocols rely only on first-hand information with optional second-hand information. CORE [11] proposed by P. Michiardi and CONFIDANT [12] proposed by Buchegger and Le Boudec fall into this category. Some basic problems with this approach of global reputation systems are:

- Every node has to maintain  $O(n)$  reputation information where  $n$  is number of nodes in network.
- Extra traffic generation in reputation exchange.
- Extra computation in accepting indirect reputation information (secondhand information) esp. Bayesian Estimation.
- Security issues in reputation exchange such as reputation data packets can be modified.

CONFIDANT detects misbehaving nodes by means of observation or by ALARM signals from neighborhood. It aggressively

informs nodes in neighborhood about misbehavior of the malicious node. The weightage of ALARM warning signal depends upon the level of trust of receiving node about the sending node. In addition, it uses bayesian estimation for various measures and calculation of trust and reputation and thus, the IDS becomes complex. CONFIDANT is vulnerable to false accusations if trusted nodes lie or if several liars collude [15].

CORE [11] proposed by P. Michiardi et. al. uses a mechanism to enforce node cooperation in MANETs. In this mechanism, reputation is a measure of someone's contribution to network operations. Members that have a good reputation can use available resources while members with a bad reputation cannot, because they refused to cooperate earlier and are gradually excluded from the community. CORE defines three types of reputation:

- Subjective reputation is a reputation value which is locally calculated based on direct observation.
- Indirect reputation is second hand reputation information which is established by other nodes.
- Functional reputation is related to a certain function, where each function is given a weight as to its importance. For example, data packet forwarding may be deemed to be more important than forwarding packets with route information, so data packet forwarding will be given greater weight in the reputation calculations.

CORE reputation values range from positive (+1), through null (0), to negative (-1). CORE suffers from the problem of unwanted consequence of good reputation, where a good node may even wish to decrease its reputation by behaving badly to prevent its resources being over-used. The CORE mechanism assumes that every node will use the same reputation calculations and will also assign the same weights to the same functions. This is a potentially inappropriate assumption in heterogeneous adhoc networks, where devices with different capabilities and roles are likely to place different levels of importance on different functions depending upon CPU usage, battery usage etc. One can take advantage of this situation and may perform only those functions which have higher preferences in calculating reputation.

The second type of IDS may be categorized as *local systems*. They solely depend upon the first hand observation of their neighbors for reputation maintenance. OCEAN [13] by Bansal and Baker falls into this category. In these systems, nodes make routing decisions based only on direct observations of their neighbor nodes. This eliminates most of the trust manager complexity, but, doesn't fit well to a highly mobile adhoc network. In such a network, it may be difficult for the reputation upgrading process to cope up with the node mobility and it might not be appropriate to depend solely upon personal observation. Also, using secondhand information can significantly accelerate the detection and subsequent isolation of malicious nodes in MANETs [16].

## IV. SYSTEM OVERVIEW

As stated earlier [17], the system design is based on the reputation paradigm and possesses a *semi-distributed* nature in terms of the reputation exchange mechanism.

The term semi-distributed is used in the system observation context, which is neither restricted to the observing node nor immediately propagated to the whole network as is the case in true distributed systems. The system design has been kept simple keeping in mind the amount of traffic already in the network and constraints such as the critical amount of battery and computational power that individual nodes possess. The system runs on every node in the network and consists of the following modules:

### A. Monitor

In wireless networks, acknowledgements are often provided at no cost, either as an existing standard part of the MAC protocol in use (such as the link-layer acknowledgement frame defined by IEEE 802.11) or by a "passive acknowledgement" (in which, a node confirms receipt at another node by overhearing the transmission from sender). The Monitor holds the responsibility of monitoring activities in the neighborhood using PACKs (Passive ACKnowledgements) which have been provided as a feature in the DSR protocol specifications [4] as promiscuous mode.

Every node registers all the data packets sent by it to its next hop neighbor and on overhearing packets in promiscuous mode, it matches those against packets registered in the queue. These packets are considered as PACKs only if both of the following two tests succeed:

- The source address, destination address, protocol, identification and fragment offset fields in the IP header of the two packets must match, and
- If either packet contains a DSR source route header, both packets must contain one and the value in the 'segments left' field in the DSR source route header of the new packet must be less than that in the first packet.

A crucial new parameter introduced in our system is the *timing window* that is a fixed time interval. After each timing window, nodes make a log of number of packets for which they have not received acknowledgment in the form of PACK and communicate this information to the reputation system. In existing reputation systems, every packet is kept waiting for its PACK for a fixed time interval. In contrast, we use the concept of timing window, which gives us the flexibility of checking timeout on fixed intervals rather than checking it on the basis of each individual packet's timeout. Monitor maintains a log of activity of next neighbor for each window and sends it to the reputation manager. Depending upon its cooperation, performance and current environment conditions, reputation system updates the nodes' reputation. With the help of Timing Window, the system also takes into consideration *congestion state* of nodes, which shall be explained in next subsection.

## B. Reputation System

Reputation system module assigns and maintains reputation of different nodes as a numeric value with a lower limit of 0 and upper limit equal to the value of *MaliciousThreshold*. Reputation of any node can change by three means, as shown in Figure 1:

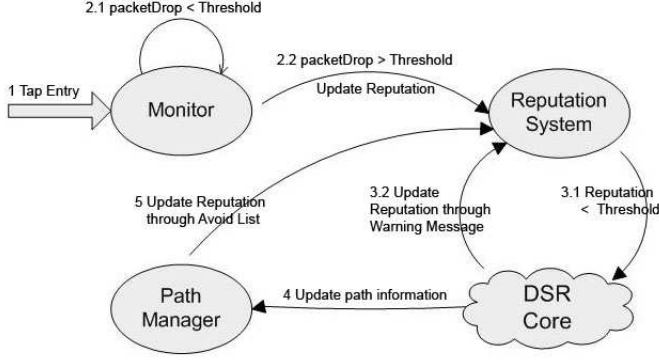


Fig. 1. The System Behavior

- By Self observation
- WARNING Message, issued by neighboring nodes
- Avoid List, appended to the RREQ/RREP header

All three means of reputation change have some associated reputation weights with them, giving maximum weightage to self observation. The reputation is updated after every timing window and the information is communicated in a sporadic way by means of avoid lists thereby avoiding much of network overhead. The concept of *avoid list* is inherited from OCEAN [13]. It facilitates easy communication among nodes by putting their malicious node list in the RREQ header. This helps in reducing the extra network traffic which would otherwise be generated while communicating this information among the peers. A node may be tagged as *normal*, *suspicious* or *malicious* depending on the reputation value associated with it. Every new node starts with a normal reputation value of zero and this reputation value may be lowered by degrading its performance or it may be incremented through the *positive appraisal* feature on normal behavior. To add to the robustness and performance of our system, it is ensured that absolute value of the negative decrement awarded is larger than the positive appraisal. However at no point should the reputation of node go above zero to prevent the kind of attacks, where a node first gains positive reputation but later on depicts a malicious behavior, thereby bringing its reputation value back to the normal range. It also avoids the peculiar situation where a node may end up exhausting all its crucial resources in routing extra traffic faced due to the popularity gained by earning positive reputation. After each timing window, reputation system receives activity log of next hop neighbor from monitor with number of packets for which it has not received PACK, which are classified as *missing* or *dropped* packets. The number of missing packets is then compared with the *MaliciousDropThreshold* and if it is comparatively

lesser, then the reputation manager gives positive performance appraisal otherwise a negative one. Unlike existing systems our system does not have a rigid *MaliciousDropThreshold*, we introduce the concept of *congestion parameter*, which is given as:

$$\text{Congestion Parameter} = \frac{\text{Current queue length}}{\text{Total queue length}} \quad (1)$$

With the assumption that the next node is also in same congestion state as the node in contention. Misbehavior drop threshold, that is the allowed number of packet drops in a timing window is dynamically decided as:

$$\text{MaliciousDropThreshold} = \text{MaxPacketRate} \times \text{CongestionParameter} \times \text{Window} \quad (2)$$

Whenever a new node is categorized as malicious, a warning message is spread only to its immediate neighborhood, thus protecting the network flooding with reputation update messages. This can be understood from the Figure 2:

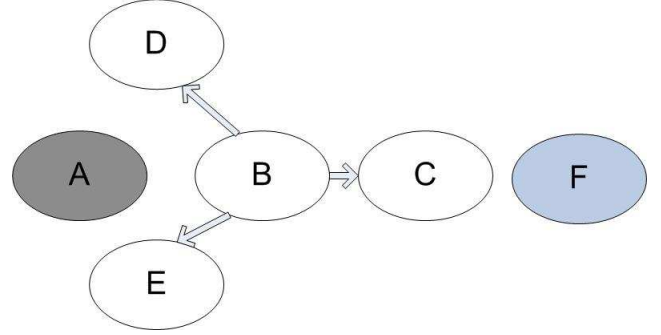


Fig. 2. A typical network scenario

If Node B categorizes A as malicious, a Warning Message is spread to all immediate neighbor nodes: C, D and E (not to F). Nodes C, D and E on receiving a warning message decrease the reputation of node A, against which the message was originally published. Lastly, another mode of reputation updating is by means of an avoid list [16]. During the route discovery phase the RREQ sending node puts its malicious node list in RREQ packet header and initiates discovery. When a node receives a route request packet it decreases the reputation of all those nodes quoted in the avoid list by a predefined weight. The node also appends its own malicious node list in the header and then forwards the route request packet.

In order to deal with the attacks on a typical reputation system, like those of ‘Collusion of liars’ and ‘false warning messages’, the system has a policy that *nodes can be categorized malicious only by self-observation*. It helps in nullifying attacks of the aforementioned types as the false warning messages spread by nodes can only decrease reputation of the victim nodes to a certain extent, termed as *suspicious threshold*. Once this threshold is reached, the system solely depends upon self observation for making the final decision. Warning messages and avoid list are only effective above the suspicious threshold.

Whenever a node's reputation is in the suspicious category and a deciding node receives a new warning message or an avoid list appearance for the previous one, the system performs a *knock test*. Knock test is a unique test designed for checking the authenticity of a node against whom the deciding node constantly receives such information. For instance, if node A falls into the suspicious category and node B receives another warning message or an avoid list appearance corresponding to node A, the deciding node B performs a knock test over A, explained later in the protocol description section.

### C. Path Manager

The path manager performs trivial path management functions in collaboration with DSR core. Path ranking is done according to path priority formula (3). Updating path-cache on various events such as those when new nodes are declared as malicious or a malicious node is taken back in network; taking decision on receiving route request or traffic from malicious nodes are a few responsibilities of the path manager. Concept of avoid list has been added to path manager, which is a list of malicious nodes that a certain node possesses and is appended to the RREQ header whenever a route request is issued by some node. Nodes which find themselves in avoid list do not process the packet and may simply drop it. During RREP, only a path with clean nodes is preferred over those containing suspicious nodes and malicious nodes. Replies from such nodes are also dropped and nodes do not process request and/or forward data packets from such nodes. If during traffic flow, a new node is declared malicious, then, all paths containing it are deleted from route cache and a route error is generated, stating that their link to the destination node has been broken. Neighbors, after receiving a route error, clear the activity log of the node which generates a route error from the current timing window. The following function may be used to decide the path priority if need arises:

$$\text{Path Priority} \propto \frac{1}{|\text{Min reputation of Node in path}| \times \text{no of hops}} \quad (3)$$

### D. Redemption And Fading

*Redemption* and *Fading* are introduced in our design to allow nodes previously considered malicious to become a part of the network again. MANETs run on cooperation and collaboration of peer nodes and no one gets benefited without cooperating with each other. Knock test is crucial for nodes in suspicious category and node may fail this test due to various reasons like transient link failures, congestion or resetting of the network interface etc. and once they fail this test, they are declared as malicious. To account for these problems, our system uses the *fading* mechanism. After a certain inactivity period the reputation of a node is improved by a certain predefined fading rate and finally the node is moved from the malicious list to middle of suspicious category. But, the node is not given *neutral rating* [18] so that if the node again misbehaves then it is immediately put in malicious list and all transactions through that node are blocked. Here, inactivity

period means no appearance in any WARNING messages or avoid list.

## V. PROTOCOL DESCRIPTION

This section entails a discussion of the actual working of the system and provides the flow for various activities at different types of nodes. Following algorithms give a concise idea of the route discovery phase, monitoring mode and knock test feature of our system as discussed in earlier sections.

### SENDER

- 1:  $\Rightarrow$  Generate RREQ Packet
- 2:  $\Rightarrow$  Pack Malicious List in RREQ Header as Avoid List
- 3:  $\Rightarrow$  Propagate Request

### OTHER NODES

- 4: **if** (Own name present in Avoid List) **then**
- 5:      $\Rightarrow$  Drop Request
- 6: **else**
- 7:      $\Rightarrow$  Scan Avoid List
- 8:      $\Rightarrow$  Update Node's Reputation
- 9:      $\Rightarrow$  Append its own malicious list to RREQ header avoiding repetition
- 10:    **if** (Node is same as Destination in RREQ) **then**
- 11:        $\Rightarrow$  Prepare Reply
- 12:    **else**
- 13:        $\Rightarrow$  Add itself in route and propagate
- 14:    **end if**
- 15: **end if**

The above algorithm presents a node's behavior during route establishment phase. Sender of the RREQ just initiates the route discovery process with avoid list of malicious nodes packed in the RREQ packet header. The remaining nodes after receiving such requests process the avoid list attached in the received RREQ header. If a matching entry is found for their own name in the list, the node drops the request. Otherwise, the reputation of the other nodes present in the avoid list is updated. If the receiving node is the destination for which the RREQ has been sent, then it prepares a route reply else it appends its own malicious list in the header to the existing avoid list avoiding repetitions and propagates the route request.

### MONITOR MODE

Self Observation-

- 1: **if** (Performance is below normalThreshold) **then**
- 2:      $\Rightarrow$  Negative reputation update
- 3: **else**
- 4:      $\Rightarrow$  Positive reputation update
- 5:    **if** (reputation is above 0) **then**
- 6:        $\Rightarrow$  SET reputation = 0
- 7:    **end if**
- 8: **end if**

### WARNING MESSAGE PROPAGATION

```

9: if (WARNING_MSG && NEIGHBOR) then
10:   if (Reputation below Suspicious Threshold) then
11:      $\Rightarrow$  Perform Knock Test
12:     if (Knock Test is Passed) then
13:        $\Rightarrow$  Assign normal reputation
14:     else
15:        $\Rightarrow$  declare as Malicious
16:        $\Rightarrow$  spread Warning Message
17:     end if
18:   end if
19: else
20:    $\Rightarrow$  decrease reputation
21: end if

```

The system in monitoring mode has three ways of gathering information for reputation updation:

- Self Observation
- Warning Message
- Avoid List

Some observations just monitor the neighbor with the help of PACK. If the performance lies below the suspicious threshold, then a negative reputation update is performed over the node in consideration, otherwise a positive appraisal is given. Warning messages are only processed if they are for immediate neighbors. If the reputation of a node under consideration is below the suspicious threshold, then the knock test is performed. Otherwise, the reputation is decreased linearly. Table 2 contains actual values of these constant parameters used during system simulation.

### *KNOCK TEST*

```

1:  $\Rightarrow$  Identify target Node
2:  $\Rightarrow$  Generate fake data packet with route via target node
3:  $\Rightarrow$  Send packet to target node and wait for its PACK
4: if (PACK is found) then
5:    $\Rightarrow$  test Passed
6:    $\Rightarrow$  Set reputation to default
7: else
8:    $\Rightarrow$  test Failed
9:    $\Rightarrow$  Declare node as malicious and broadcast Warning
      message
10: end if

```

Knock test is designed specifically for immediate neighbors to test whether a particular node is malicious or not and is only performed on nodes in suspicious state. In this test a dummy data packet with time to live (TTL) equal to 2 is sent to a node in question via last known route through that node.

The sender node overhears traffic of the node in question in promiscuous mode. If the node on which knock test is being performed successfully forwards the test packet to next hop then its reputation is set to default. In case it fails, then it is immediately put into malicious category and a warning message is broadcasted in the immediate neighborhood. If in

case, the dummy packet is genuinely dropped because of bad channel conditions the node may be classified as malicious. However, it still has an opportunity to become a part of network again through redemption and fading mechanism, as explained earlier. This is done because the system only trusts first hand information for putting a node into malicious category, thus, giving self observation the highest weightage. The weightage assigned to warning message and avoid list citation is comparatively less than self observation.

## VI. IMPLEMENTATION/SIMULATION

This section first describes the simulation environment and then we compare the throughput of our system in the presence of malicious nodes against a defenseless DSR protocol. The network simulator ns2 (version 2.29) [19] was used to run the simulations. Mobility of nodes is characterized by a mobility model, speed and ‘pause time’. The random waypoint model is selected as a mobility model in a  $1000 \times 1000 \text{ m}^2$  rectangular field. Using this mobility model, each movement is a straight line between a start and an arrival point, covered at a constant speed which is a uniform distribution, between 0 and  $10 \text{ m.s}^{-1}$  for each movement. The pause time is the time period between two consecutive movements. Thus, the higher the pause time, lesser is the node’s mobility. We have used 5 different pause times: 0, 100, 300, 600 and 900 seconds.

There are two setups having a total of 10 and 20 nodes, with number of malicious nodes between 10 to 100%. We use maximum 5 and 10 CBR (Constant Bit Rate) connections for 10 and 20 nodes respectively, sending 64 bytes packets with a  $4 \text{ pkts.s}^{-1}$  sending rate. The bandwidth is  $2 \text{ Mb.s}^{-1}$ . The Medium Access Control (MAC) protocol used is IEEE 802.11. The malicious nodes are of the following nature: dropping an average of 99% of the CBR-connection packets (data packets). The dropping decision is taken depending upon a number generated at random. We assume that malicious nodes do not drop the DSR routing packets like route request, route reply or error as they always want to be part of network. A malicious node dropping all the packets is comparatively less dangerous for the MANET because in that case, it would drop all packets including routing packets. Following which, they would never be able to include themselves in any the communication routes. The fixed parameters for the simulation are listed in Table 1.

**Table 1: Fixed Parameters**

| Parameter        | Level                                                     |
|------------------|-----------------------------------------------------------|
| Area             | $1000 \text{ m} \times 1000 \text{ m}$                    |
| Speed            | uniformly distributed between 0 and $10 \text{ m.s}^{-1}$ |
| Radio Range      | 250 m                                                     |
| Placement        | uniform                                                   |
| Movement         | random waypoint model                                     |
| MAC              | 802.11                                                    |
| Sending capacity | 2 Mbps                                                    |
| Application      | CBR                                                       |
| Packet size      | 64 B                                                      |
| Simulation time  | 900 s                                                     |

Thus, if no route has been established containing these nodes, they would never be able to drop any data packet sent to them either. As a consequence, they would not affect the throughput of whole network. For evaluating the performance of our system, we account for the *Packet Delivery Ratio* and *Routing Overhead* metrics. Packet delivery ratio is calculated as the ratio of data packets received to data packets sent. For routing overhead we have taken a ratio of number of control packets generated (request, reply and error) to the number of data packets sent thus, being basically a cost v/s. gain ratio. The routing overhead ratio gives us the approximate number of control packets for each data packet sent which should not be significantly greater as compared with that of normal DSR. Throughput refers to the actual measured performance of the system when the delay is considered. In the simulation results, the metrics of throughput are related to the average value per node. Finally, the average delay shows the average one-way latency observed between transmitting and receiving a packet. Unless otherwise specified, the experiments are repeated ten times in all cases with varying random seed. The seed influences the placement and movement of the nodes. The radio range, sending capacity, and MAC have been chosen to represent a typical adhoc mobile node; the speed is uniformly distributed between 0 and 10 m/s to represent speed of user in fixed location, walking or running. The simulation time is chosen to be long enough to potentially roam the whole area and is set to 900 seconds. The system was deployed and tested with following values of constant parameters

**Table 2: Values of Constant Parameters**

| Constant                        | Value              |
|---------------------------------|--------------------|
| Neutral Rating                  | 0                  |
| Suspicious Threshold            | -35                |
| Malicious Threshold             | -50                |
| Window Size                     | 1 second (Default) |
| Self Observation Weightage      | -5                 |
| Warning Message Weightage       | -2                 |
| Avoid List Appearance Weightage | -1                 |
| Inactivity Timeout Period       | 20 seconds         |

Finally, CBR has been chosen for generating the traffic. The scenario and traffic connections have been randomly generated using the cbrgen and setdest utility from CMU's Monarch project [20].

## VII. RESULTS AND PERFORMANCE EVALUATION

Results for Packet Delivery Ratio are shown in the Figure 3. The system performs better than normal DSR comparing results taken after average over 10 iterations with different pause times. The system performance is significantly better than normal DSR when percentage of malicious nodes is less than 40. After which, it starts to deteriorate and significantly falls after 70%. But, in case 70% or more nodes are malicious we can simply discard the network as it is no longer of significance. There is no need to establish trust relationship and links among the nodes when 7 out of 10 are known to be malicious.

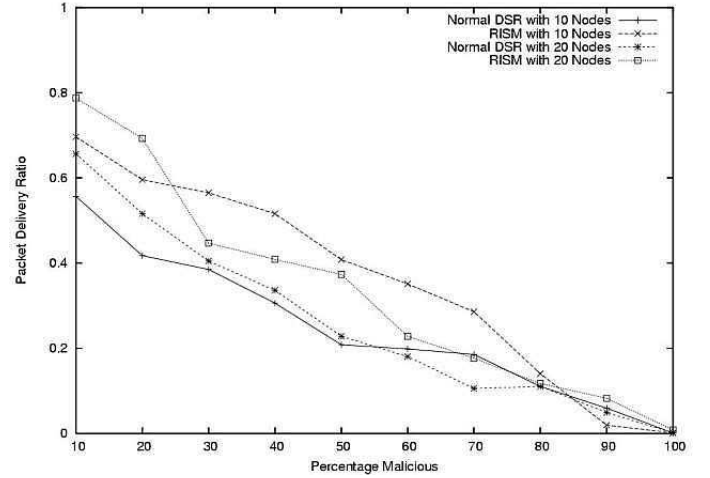


Fig. 3. Packet Delivery Ratio Comparison

Figure 4 shows routing overhead of our system protocol compared with normal DSR. Number of control messages in the network are significant, as more are the number of packets, more is the time spent in establishing routes and lesser is the number of data packets sent. Our system performs better than normal DSR without much extra routing overhead. This extra routing overhead is generated because whenever a new node is declared as malicious, a route error is generated and the link is broadcast as broken. After which, some more time is consumed to establish a new link. This is crucial to the IDS performance.

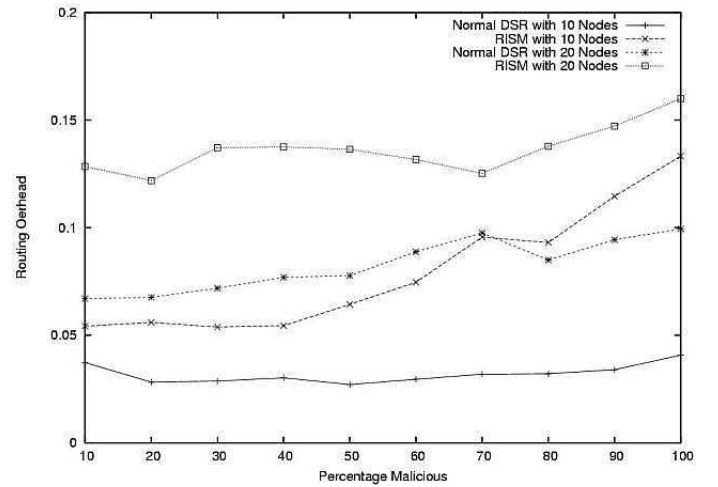


Fig. 4. Routing Overhead Comparison

The Figure 5 illustrates fine tuning of the system with respect to sliding window size. Window size is a crucial parameter for the system because of its role in deciding the system performance. As depicted by the figure, in most cases, the window size 1.25 seconds scenario delivers optimal packet delivery ratio as compared to other scenarios where window size is of 0.50, 0.75, 1.00, 1.50 and 1.75 seconds. From the figure, one can infer that for a small window size, the system

is too busy in various book-keeping tasks for monitoring and reputation updating. For a larger window size, the system response gets too slow. Hence, the time to identify malicious nodes increases and accordingly does the number of packet drops. Therefore, overall performance of system deteriorates in terms of packet delivery ratio.

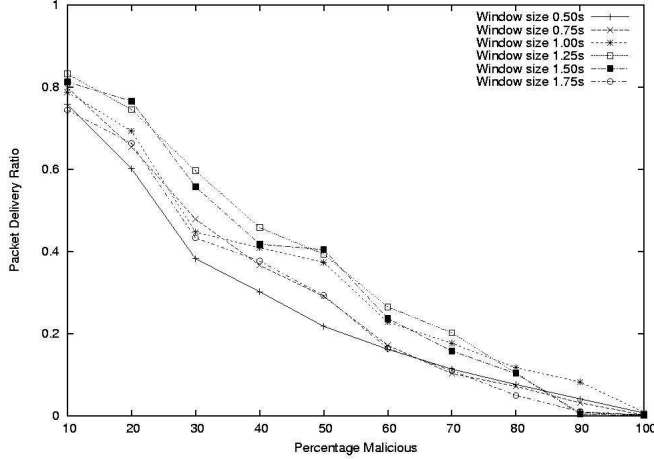


Fig. 5. System performance for various window sizes

The Figure 6 shows packet delivery ratio values for the system against pause times of 0, 100, 300, 600 and 900 seconds.

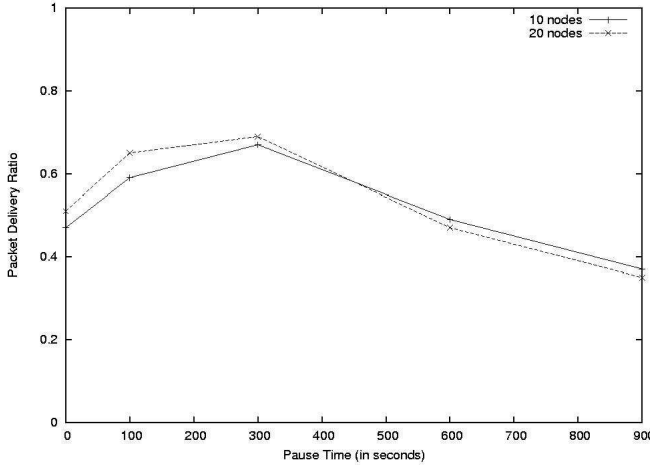


Fig. 6. System performance against Pause times

In a highly mobile scenario such as one with pause time of 0 seconds the system performance decreases as most of the time the system has to cope up with the mobility of nodes and tasks like updating route cache, discovering & establishing routes etc. Likewise, in a static network scenario with pause times of 600 and 900 seconds, where the system does not have many choices in terms of clean routes, once these nodes get identified, the system performance also degrades.

As depicted by the Figure 7, the performance is optimal for a scenario with window size of 1.25 seconds in terms of

routing overhead. Although, the difference between various scenarios presented is not very significant, but it is crucial for system performance to optimize this value.

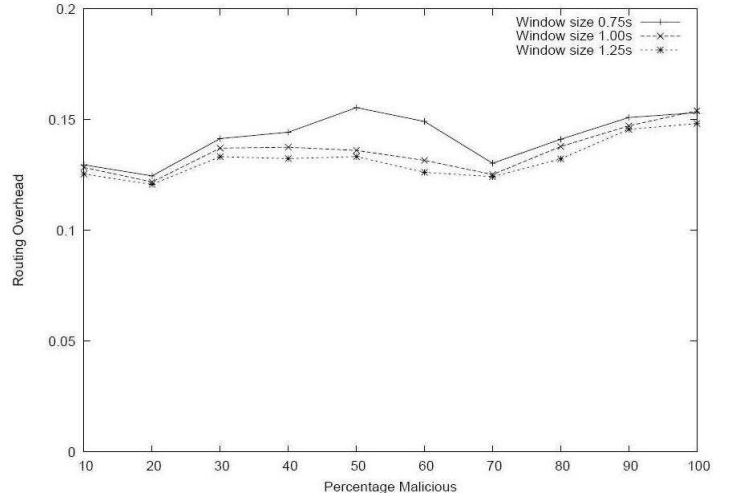


Fig. 7. System routing overhead for various window sizes

In addition, the system also performs optimally in terms of the routing overhead incurred. Thus, the proposed solution is able to strengthen the defense of DSR protocol without incurring much of overhead.

## VIII. CONCLUSION AND FUTURE PERSPECTIVE

Mobile adhoc networks have a number of significant security issues which cannot be solved alone by simple IDS. In this paper, we have critically examined the existing systems and outlined their strength and shortcomings. We have opted a semi-distributed approach for our system in terms of mode of information propagation among nodes. The goal was to design a system incorporating the best traits of all existing systems without incurring extra routing overhead. Congestion parameter, Knock test and Timing window are some new concepts that have been introduced in this system. Detailed simulations carried out over our system using ns2 for performance evaluation have contributed significantly to some crucial design issues. Optimal values of the parameters used are obtained and critically examined for efficient performance of the system. However, some additional study is required for evaluating the adequacy and importance of congestion parameter. The system performance can also be judged by interchanging the values of weightage assigned to self observation with that of other reputation update modes such as warning message and avoid list citations. It is our belief that some interesting results are bound to come with such studies which shall justify the system design in its current stage.

## REFERENCES

- [1] C. E. Perkins, P. Bhagwat: "Highly Dynamic Destination-Sequenced Distance Vector (DSDV) for Mobile Computers" *Proc. of the SIGCOMM 1994 Conference on Communications Architectures, Protocols and Applications*, pp 234-244, Aug (1994).



- [2] S. Murthy, J.J. Garcia-Luna-Aveces: "A Routing Protocol for Packet Radio Networks", *Proc. ACM International Conference on Mobile Computing and Networking*, pp. 86-95, November, (1995).
- [3] Ching-Chuan Chiang, Hsiao-Kuang Wu, Winston Liy, Mario Gerla: "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel", *IEEE Singapore International Conference on Networks, (SICON'97)*, pp. 197-211, Singapore, 16.-17. April (1997).
- [4] David B. Johnson, David A. Maltz, and Josh Broch, "DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks" In *Ad Hoc Networking*, edited by Charles E. Perkins, chapter 5, pages 139-172. Addison-Wesley, (2001).
- [5] Charles E. Perkins and Elizabeth M. Royer: "Ad-hoc On-Demand Distance Vector Routing" in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, pp. 90-100, February (1999).
- [6] V. Park, S. Corson: "Temporally-Ordered Routing Algorithm (TORA)" VERSION 1 Internet Draft, draft-ietf-manet-tora-spec- 03.txt, June (2001).
- [7] Chai-Keong Toh: "A Novel Distributed Routing Protocol To Support Ad hoc Mobile Computing", *Proc. IEEE 15th Annual International Phoenix Conference on Computers and Communications*, IEEE IPCCC 1996, 27 March-29, Phoenix, AZ, USA, pp. 480-486 (1996).
- [8] R. Dube, C. D. Rais, K. Wang and S. K. Tripathi: "Signal Stability based adaptive routing (SSR alt SSA) for ad hoc mobile networks", *IEEE Personal Communication*, Feb. (1997).
- [9] Zygmunt J. Haas, Marc R. Pearlman, Prince Samar: "The Zone Routing Protocol (ZRP) for Ad Hoc Networks", Internet Draft, <http://www.ietf.org/proceedings/02nov/I-D/draft-ietf-manet-zone-zrp-04.txt>, work in progress, July (2002).
- [10] Sonja Buchegger, Cedric Tissieres, Jean-Yves Le Boudec, "A Test-Bed for Misbehavior Detection in Mobile Ad-hoc Networks How Much Can Watchdogs Really Do?," *Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'04)*, pp. 102-111, (2004).
- [11] P. Michiardi, R. Molva, "Core: A Collaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks", *Institut Eurecom Research Report RR-02-062 - December (2001)*.
- [12] Sonja Buchegger and Jean-Yves Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes:Fairness In Dynamic Ad-hoc NeTworks" *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Lausanne, June(2002).
- [13] Sorav Bansal and Mary Baker, "Observation based cooperation enforcement in ad hoc networks" Technical Report, Stanford University, arXiv:cs.NI/0307012 v2 6 Jul (2003).
- [14] P. Resnick and R. Zeckhauser. "Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system" In M. R. Baye, editor, *The Economics of the Internet and E-Commerce*, volume 11 of *Advances in Applied Microeconomics*. Amsterdam, Elsevier Science, (2002).
- [15] Sonja Buchegger, Jean-Yves Le Boudec: "Coping with False Accusations in Misbehavior Reputation Systems for Mobile Ad-hoc Networks" EPFL Technical Report IC/2003/31 (2003).
- [16] S. Buchegger and J.-Y. Le Boudec, "The effect of rumor spreading in reputation systems for mobile ad-hoc networks" *Proc. WiOpt'03(Modeling and Optimization in Mobile Ad Hoc and Wireless Networks)*, (2003).
- [17] Animesh K. Trivedi, Rishi Kapoor, Rajan Arora, Sudip Sanyal and Sugata Snayal: "RISM - Reputation Based Intrusion Detection System for Mobile Adhoc Networks", accepted in CODEC'06, Kolkata, India (<http://www.irpel.org/phpfiles/codec-06.php>) to be held in Dec. (2006).
- [18] P. Yau and C. J. Mitchell, "Reputation methods for routing security for mobile ad hoc networks", in *Proceedings of SympoTIC '03*, Joint IST Workshop on Mobile Future and Symposium on Trends in Communications, Bratislava, Slovakia, October (2003).
- [19] NS Home Page: <http://www.isi.edu/nsnam/ns/>
- [20] "CMU Monarch Project web site:" <http://www.monarch.cs.cmu.edu/>
- [21] Sonja Buchegger and Jean-Yves Le Boudec, "Self-Policing Mobile Ad-Hoc Networks by Reputation Systems" *IEEE Communication Magazine*, vol. 43, num. 7, p. 101(2005).
- [22] Sergio Marti, T.J. Giuli, Kevin Lai, and Mary Baker, "Mitigating routing misbehavior in mobile ad hoc networks" *Proceedings of the 6th annual international conference on Mobile computing and networking* Boston, Massachusetts. Pages: 255 - 265 (2000).

**Animesh Kumar Trivedi** is a senior undergraduate student of Information Technology at the Indian Institute of Information Technology, Allahabad, India. His research interests include Security issues in Wireless Computer Networks, Computer Architecture and Distributed Systems. Further details about him can be had from <http://animesh.trivedi.googlepages.com/>

**Rajan Arora** is a senior undergraduate student of Information Technology at the Indian Institute of Information Technology, Allahabad, India. His research interests include Wireless Computer networks, Cryptography, Network Security and Distributed systems. Further details about him can be had from <http://arorarajan.googlepages.com/>

**Rishi Kapoor** is a senior undergraduate student at the Indian Institute of Information Technology, Allahabad, India pursuing his undergraduation in Information Technology. His research interests include Computer and Wireless networks, Network security and Databases. Further details about him can be had from [http://profile.iiita.ac.in/rkapoor\\_b03/](http://profile.iiita.ac.in/rkapoor_b03/)

**Sudip Sanyal** is an Associate Professor at the Indian Institute of Information Technology, Allahabad, India. He received his M.S. and Ph.D. degree from the Banaras Hindu University at Varanasi, India. His current activities lie in fields of Computer networks, Software Engineering,

Parallel Computing and Natural language Processing. He has published numerous papers in various national and international journals and attended many conferences.

**Sugata Sanyal** is in the Faculty of the Tata Institute of Fundamental Research, India. He received his Ph.D. degree from Mumbai University, India, M.Tech from IIT, Kharagpur, India and B.E. from Jadavpur University, India. His current research interests include security in wireless and mobile ad hoc networks, distributed processing, and scheduling techniques. He has published numerous papers in national and international journals and attended many conferences. He is in the editorial board of three International Journals. He is co-recipient of Vividhlaxi Audyogik Samsodhan Vikas Kendra Award (VASVIK) for Electrical and Electronics Science and Technologies (combined) for the year 1985. He was a Visiting Professor in the Department of Electrical and Computer Engineering and Computer Science in the University of Cincinnati, Ohio, USA in 2003. He delivered a series of lectures and also interacted with the Research Scholars in the area of Network Security in USA, in University of Cincinnati, University of Iowa, Iowa State University and Oklahoma State University. He has been an Honorary Member of Technical Board in UTI (Unit Trust of India) and SIDBI (Small Industries Development Bank of India). He has also acted as a consultant to a number of leading industrial houses in India. More information about his activities is available at <http://www.tifr.res.in/~sanyal>.